ISSN:0975-9646

K.Venkata Ganesh Babu et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (6) , 2011, 2482-2484.

Dynamic Bandwidth for Minimum Delay Peer-to-peer Streaming

K. Venkata Ganesh Babu (M.Tech)
Kakinada Institute of Engg & Tech, Tallarevu

K. Ravi Kumar M.Tech
Assistant Prfessor, Kakinada Institue of Engg & Tech

Abstract

In the peer to peer network the high bandwidth demand for uploading multimedia applications. The uploading bandwidth of individual peers to distribute content at low server cost. The peer to peer bandwidth sharing design is very efficient for bandwidth sensitive applications. The uploading bandwidth of peer cannot be utilized the piece of content until it completes the download of the content . In this paper, we propose a fully dynamic algorithm to achieve optimal peer selection and streaming rate allocation, which minimizes peer -to-peer latencies in the streaming sessions. We design this efficient dynamic algorithm based on the solution to a linear optimization model, which optimizes towards a latency-related objective to decide the best streaming rates among peers. Combining this optimal peer selection algorithm with our coding scheme based on rate less codes, we obtain a complete, fully decentralized minimum-delay peer-to-peer streaming scheme.

Keywords: peer selection, peer to peer streaming, minimizing delay, media streaming

I. Introduction

Multiple peers from different overlays are in conflict with one another, competing for limited upload bandwidth at the same streaming server or upstream peer in the network. In this case, the allocation of such upload bandwidth needs to be meticulously mediated with appropriate strategies, such that the streaming rate requirement of each overlay is satisfied at all their participating peers. It would be best if, at the same time, fairness can be achieved across different overlays, and costs of streaming (e.g., latencies) can be minimized. Most existing P2P streaming studies focus on a single overlay, neglecting the conflict scenarios among coexisting overlays with respect to available bandwidth. Limited bandwidth capacities in peer-to-peer networks pose a significant technical challenge in peer-to-peer media streaming. As nodes in peer-to-peer networks reside at the edge of the Internet, they usually have limited availability of upload and download capacities. In addition, due to peer heterogeneity, the available peer node bandwidth may differ by at least an order of magnitude. For delay-sensitive media streaming applications, the typical streaming bit rates in modern streaming codes must be accommodated for all the peers in a streaming session, in order to ensure their uninterrupted streaming playback. Therefore, it is typical for a peer node to parallel download from multiple upstream peers, in order to improve the overall bandwidth availability. In this case, a critical question arises: What is the best way for the peer nodes to select the upstream peers and allocate the streaming rates among the selected peers, such that a specified aggregate streaming bit rate is satisfied and the end-to-end latencies are minimized at the receivers. It is a nontrivial problem to obtain a feasible *peer selection and streaming rate allocation* strategy which guarantees all requesting peers can acquire the streaming bit rate of the session, not to mention that which minimizes end-to-end latencies. When the streaming rates from selected upstream peers have been optimally allocated, the next question to answer is how to assign the media contents to be delivered along each link. In the constructed mesh streaming topology featuring parallel retrievals, there are always risks that the same contents may be unnecessarily supplied by multiple upstream peers. Therefore, the peer nodes need to reconcile differences among the sets of content segments they hold. A *content assignment* scheme, which schedules which content segment to retrieve from which upstream peer, needs to be designed to minimize the delivery redundancy. we have proposed an efficient coding scheme based on rateless codes to address the later problem of delivery redundancy and reconciliation. Based on the loss resilience and "ratelessness" properties of rateless codes, our scheme provides excellent resilience to network dynamics, and guarantees that no duplicated contents exist in the network. This completely eliminates the need for set reconciliation and content assignment on the links, which otherwise involves high computation and messaging overhead [7]. Combining the optimal peer selection algorithm with the rateless code coding scheme, we are able to derive a complete resilient and optimal peer-to-peer streaming solution.

II. Dynamic Distribution Algorithm

Based on the sub gradient algorithm and primal solution recovery algorithm, we design our distributed algorithm to

solve the linear program and achieve the optimal streaming rates on the links. The distributed algorithm to be executed by link (i, j) is summarized in Table 1. In practice, we have each link (i, j) in the peer-to-peer network delegated by the receiver j, and thus the
computation tasks on all the incoming links of one peer is carried out by the peer.

The Dynamic distributed algorithm of link (i, j)
1. Choose initial Lagrangian multiplier values μt ij [0], 8(i, j) 2 A, 8t 2 T.
2. Repeat the following iteration until the sequence {μ[k]} converges to μ□ and the sequence { b f[k]} converges to b f□: 8(i, j) 2 A, 8t 2 T
1) Compute xij [k] by the distributed auction algorithm;
2) Compute ft ij [k] by the distributed Bellman-Ford algorithm;
3) Compute c ft ij [k] = k−1 k c ft ij [k − 1] + 1 k ft ij [k];
4) Update Lagrangian multiplier μt ij [k + 1] = max(0, μt ij [k] + _[k](ft ij [k] − xij [k])), where _[k] = a/(b + ck).
3. Compute the optimal multicast streaming rate zij = maxtT c ft ij . By delivering themedia contents at the computed optimal streaming rates on the links, we achieve minimum-delay peer-to-peer streaming. We further emphasize that this is actually achieved by applying our coding scheme with rateless codes, but without the complex set reconciliation and content assignment in the streaming session.

III. Performance Evolution

We first investigate the convergence speed of our distributed algorithm to obtain the optimal streaming topology. We compare the convergence speed in networks of different *network sizes* (numbers of peers in the network) and different *edge densities* (the ratio of the number of edges to the number of peers in the network). We can see that it takes around 70 iterations to converge to optimality in a network of 50 peers, and this number increases slowly to about 170 for a network of 500 peers. However, the convergence speed remains approximately the same in a fixed-sized network with different edge densities. Therefore, the slow increase of iteration numbers with network sizes does not affect the scalability of our algorithm. We further compare the convergence speeds of our algorithm to the first primal feasible solution, to the feasible solution which achieves 90% optimality as to the value of the objective function, and to the optimal solution. We observe that the convergence speed to the first primal feasible solution is usually much faster than the convergence to optimality. It can also be seen that the number of iterations needed to converge to feasibility drops quickly with the increase of the percentage of Ethernet peers in the network, which bring more abundant upload capacities. Furthermore, in order to converge to the feasible solution which achieves 90% optimality, the algorithm takes only 75% of the number of iterations required for convergence to the optimal solution. Therefore, in practice, we can obtain a feasible solution to a certain degree of the optimality in a much shorter time, when it is not always necessary to achieve the optimal solution in a realistic streaming system. We next compare our optimal peer selection algorithm with a commonly used peer selection heuristic [11, 22]. In the heuristic, a receiver distributes the streaming rates among its upstream peers in proportion to their upload capacities. We compare the end-to-end latencies at receivers in the resulting streaming topologies. The end-to-end latency at each receiver is calculated as the weighted average of the delays of flows from all its upstream peers, and the weight for each flow is the portion of the assigned streaming rate from the upstream peer in the aggregate streaming rate. The results illustrated in Fig. 6 meet our expectations. In networks of different network sizes and edge densities, our end-to-end latency minimization algorithm is able to achieve much lower latencies than the heuristic, which does not take link delay into consideration. We further notice that the denser the network is, the higher the average end-to-end latency is by the heuristic. In contrast, our optimal algorithm achieves lower latencies in denser networks. When the edge density is 4N in a network of N peers, the average end-to-end latency of the heuristic is about 1.5 times higher than that of our optimal algorithm, while this ratio becomes 2 in a network with 8N edges. For such an achievement of lower latencies in denser networks with our algorithm, we believe the reason is that there are more choices of upstream peers in a denser network and our algorithm can always find the best set of upstream peers on low delay paths. Thus, in realistic peer-to-peer streaming networks with high edge densities, the advantage of our algorithm is more evident over the commonly used heuristic.

IV. Conclusion

In this paper is to design an efficient distributed algorithm for optimal peer selection and streaming rate allocation in peer-to-peer streaming. For this purpose, we formulate the problem as a linear optimization problem, which optimizes bandwidth utilization towards minimized end-to-end latencies. Based on the efficient sub gradient solution, we develop a fully decentralized algorithm to efficiently compute the optimal streaming rates over the peer-to-peer links. We believe combining this optimal peer selection algorithm with our rateless-code coding scheme, it provides a complete solution to battle on the fundamental challenges nof peer-to-peer streaming: dynamics, reconciliation, and limited bandwidth. In the ongoing work, we are working towards a practical implementation of the minimum-delay peer-to-peer streaming system, in the real-life environment of the Internet.

References

[1] D. P. Bertsekas and D. A. Castanon. The Auction Algorithm for the Transportation Problem. *Annals of Operations Research*, 20:67–96, 1989.
[2] D. P. Bertsekas and R. Gallager. *Data Networks, 2nd Ed.* Prentice Hall, 1992.
[3] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice Hall, 1989.
[4] J. Byers, J. Considine, M. Mitzenmacher, and S. Rost. Informed Content Delivery Across Adaptive Overlay Networks. In *Proc. of ACM SIGCOMM 2002*, August 2002.
[5] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. SplitStream: High-Bandwidth Multicast in Cooperative Environments. In *Proc. of the 19th ACM Symposium on Operating Systems Principles (SOSP) 2003*, October 2003.

[6] H. Deshpande, M. Bawa, and H. Garcia-Molina. Streaming Live Media over a Peer-to-Peer Network. Technical report, Standford Database Group 2001-20, August 2001.
[7] M. Hefeeda, A. Habib, B. Botev, D. Xu, and B. Bhargava. PROMISE: Peer-to-Peer Media Streaming Using CollectCast. In *Proc. of ACM Multimedia 2003*, November 2003.
[8] J. Liu, S.G. Rao, B. Li, and H. Zhang, "Opportunities and Challenges of Peer-to-Peer Internet Video Broadcast," Proceedings of the IEEE, vol. 96, no. 1, pp. 11– 24, 2007.
[9] X. Hei, Y. Liu, and K.W. Ross, "IPTV over P2P Streaming Networks: The Mesh-Pull Approach," IEEE Comm. Mag., vol. 46, no. 2, 2008.
[10] C. Feng, B. Li, and B. Li, "Understanding the Performance Gap between Pull-based Mesh Streaming Protocols and Fundamental Limits," in Proc. of the 28th IEEE INFOCOM, 2009.
[11] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowston, and A. Singh, "SplitStream: High-Bandwidth Multicast in Coorperative Environments," in Proc. of the 19th ACM SOSP, 2003.
[12] V. Venkataraman, K. Yoshida, and P. Francis, "Chunkyspread: Heterogeneous Unstructured Treebased Peer-to-Peer Multicast," in Proc. of the 14th IEEE ICNP, 2006.
[13] N. Magharei, R. Rejaie, and Y. Guo, "Mesh or Multiple-Tree: A Comarative Study of Live P2P Streaming Approaches," in Proc. of the 26th IEEE INFOCOM, 2007.